

IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) An extensible rule-based technique for optimizing predicated code, comprising:

if-converting an abstract internal representation, wherein the if-converting comprises:

testing a condition code associated with conditional

instruction, and

writing Boolean data to a general register designated as a

destination register based on the testing, the

destination register representing a predicate;

transforming the if-converted representation to a machine representation,

wherein the transformation includes eliminating the predicate[[s]]

from the if-converted representation; and

optimizing the machine representation based on a combination of a

predetermined cover analysis and a predetermined replacement

pattern such that a redundant instruction in the machine

representation is eliminated.
2. (Canceled)
3. (Original) The technique of claim 1, the eliminating of predicates comprising:

eliminating a predicate defining instruction by interpretation.

4. (Original) The technique of claim 1, the eliminating of predicates comprising:
eliminating a guarding predicate of a safe instruction by speculation.
5. (Original) The technique of claim 1, the eliminating of predicates comprising:
eliminating a guarding predicate of an unsafe instruction by compensation.
6. (Original) The technique of claim 1, the eliminating of predicates comprising:
eliminating a guarding predicate of an unsuitable instruction by reverse if-conversion.
- 7-9. (Canceled)
10. (Currently Amended) An apparatus for optimizing predicate code, comprising:
means for if-converting an abstract internal representation, wherein the
means for if-converting comprises:
means for testing a condition code associated with
conditional instruction, and
means for writing Boolean data to a general register
designated as a destination register based on the
testing, the destination register representing a
predicate;
means for transforming the if-converted representation to machine

representation, wherein the transformation includes eliminating the
predicate[[s]] from the if-converted representation; and

means for optimizing the machine representation based on a combination
of a predetermined cover analysis and a predetermined
replacement pattern such that a redundant instruction in the
machine representation is eliminated.

11. (Canceled)
12. (Original) The apparatus of claim 10, the eliminating of predicates comprising:
means for eliminating a predicate defining instruction by interpretation.
13. (Original) The apparatus of claim 10, the eliminating of predicates comprising:
means for eliminating a guarding predicate of a safe instruction by
speculation.
14. (Original) The apparatus of claim 10, the eliminating of predicates comprising:
means for eliminating a guarding predicate of an unsafe instruction by
compensation.
15. (Original) The apparatus of claim 10, the eliminating of predicates comprising:
means for eliminating a guarding predicate of an unsuitable instruction by
reverse if-conversion.

16. (Canceled)

17. (Currently Amended) An extensible rule-based technique for optimizing predicated code, comprising:

if-converting an abstract internal representation, wherein the if-converting

comprises:

testing a condition code associated with conditional

instruction, and

writing Boolean data to a general register designated as a

destination register based on the testing, the

destination register representing a predicate;

transforming the if-converted representation to a machine representation,

wherein the transformation includes eliminating the predicate[[s]]

from the if-converted representation,

wherein ~~the eliminating of~~ the predicates, comprises at least one of

eliminating a predicate defining instruction by interpretation;

eliminating a guarding predicate of a safe instruction by
speculation;

eliminating a guarding predicate of an unsafe instruction by
compensation;

eliminating a guarding predicate of an unsuitable instruction by
reverse if-conversion; and

optimizing the machine representation based on a combination of a
predetermined cover analysis and a predetermined
replacement pattern such that a redundant instruction in the
machine representation is eliminated.

18-23. (Canceled)